FIG. 1

FIG. 2

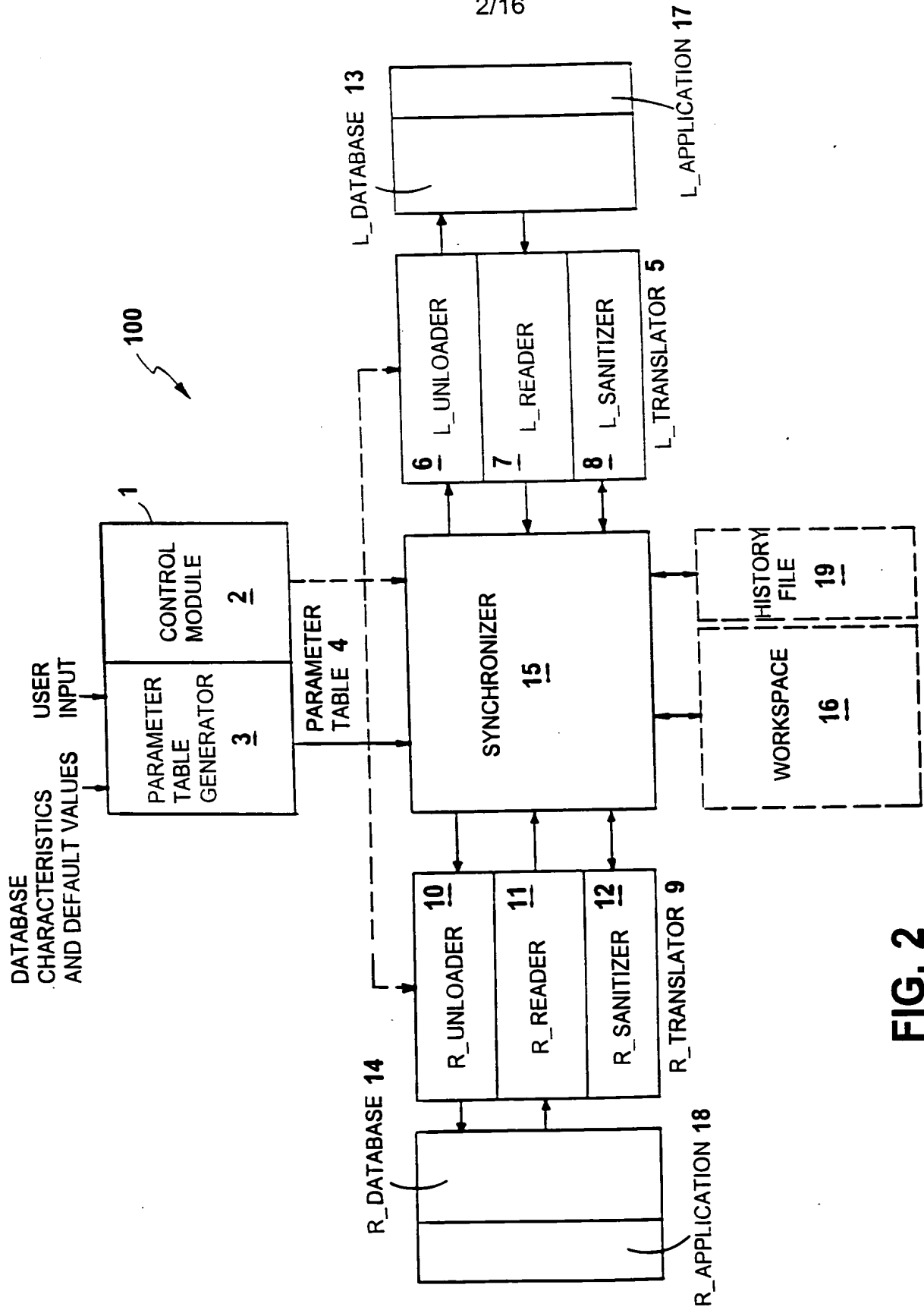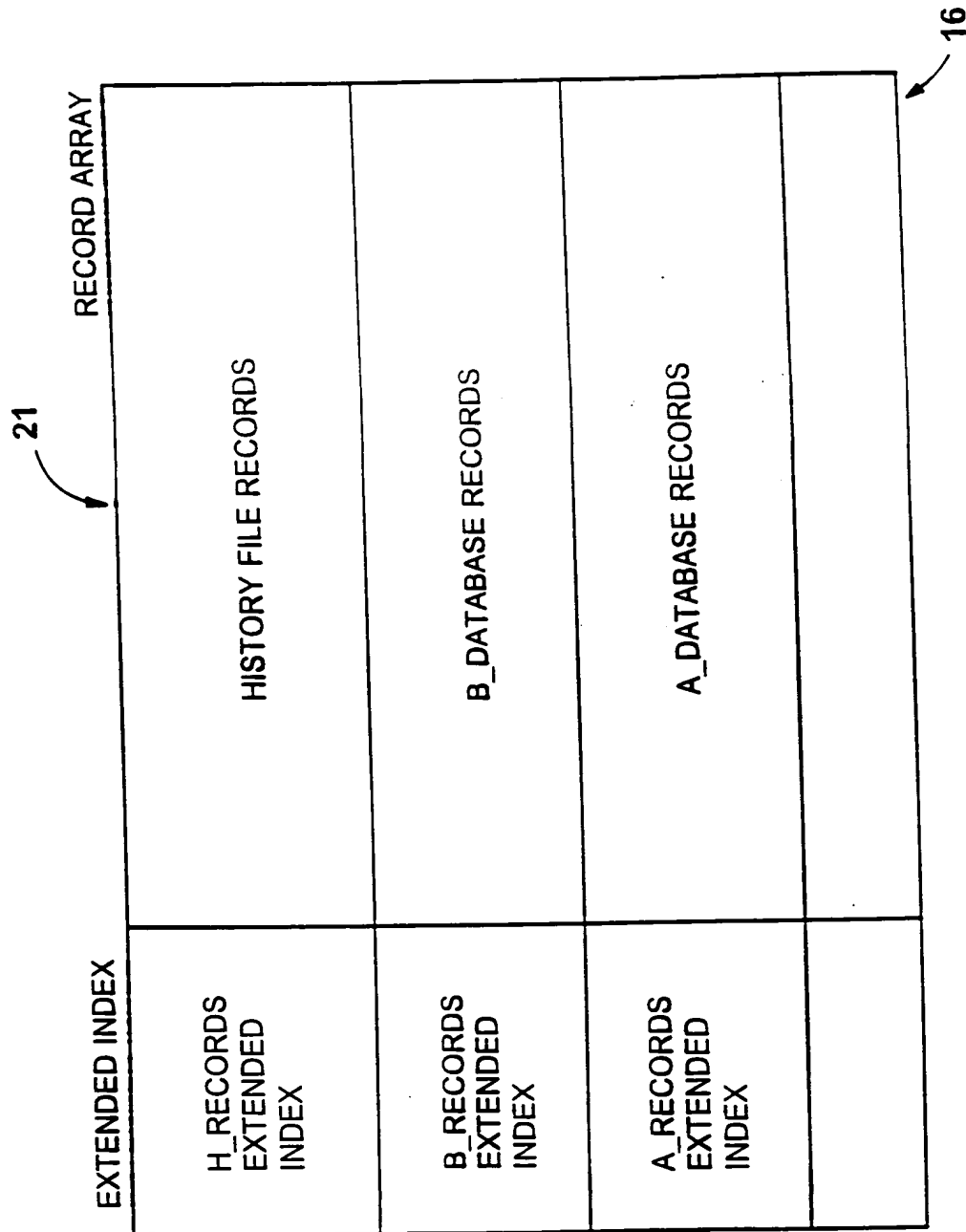| EXTENDED INDEX	RECORD ARRAY
H_RECORDS EXTENDED INDEX	HISTORY FILE RECORDS
B_RECORDS EXTENDED INDEX	B_DATABASE RECORDS
A_RECORDS EXTENDED INDEX	A_DATABASE RECORDS

21

16

FIG. 3

Pseudo Code for Translation Engine Control Module

100. INSTRUCT parameter table generator to create parameter table and initialize filter
101. INSTRUCT Synchronizer to initialize itself
102. INSTRUCT Synchronizer to LOAD the History_File into its WORKSPACE
103. INSTRUCT R_Translator to LOAD R_records from R_Database
104. INSTRUCT L_Translator to SANITIZE R_records that were just LOADED
105. INSTRUCT L_Translator to LOAD L_Records from L_Database and SEND to Synchronizer
106. INSTRUCT R_Translator to SANITIZE L_Records that were just LOADED.
107. INSTRUCT Synchronizer to do CAAR (Conflict Analysis And Resolution) on all the records in WORKSPACE.
108. INFORM user exactly what steps Synchronizer proposes to take (i.e. Adding, Changing, and Deleting records). WAIT for User.
109. IF User inputs NO, then ABORT.
110. INSTRUCT R_Translator to UNLOAD all applicable records to R_Database.
111. INSTRUCT L_Translator to UNLOAD all applicable records to L_Database.
112. INSTRUCT Synchronizer to CREATE a new History File.

FIG. 4

Pseudocode for Generating Parameter Table

{Get Input from the user}

150.   ASK user to select whether to use a filter expression

151.   IF the user selected to use a filter THEN

152.   IF a new filter to be used THEN

153.   Obtain from the user filter name

154.   Obtain filter expression

155.   STORE the current date and time in the FILTER_CHANGED_TIMESTAMP parameter

156.   Assign a unique filter ID to the filter

157.   ELSE Obtain from the user filter name

158.   retrieve the filter expression and unique filter ID

159.   IF user selects to edit the filter THEN display the filter and obtain edits

160.   SET FILTER_ID parameter to unique filter ID code of the selected filter

161.   SET USE_FILTER flag

162.   PARSE the filter expression into a filter token array

163.   END IF

164.   CREATE parameter table

FIG. 5

Advanced Settings for Phone Records

Conflict Resolution Filters

You may restrict the scope of this operation by applying a filter to MS Schedule+7.0 (chosen from the list below). Use the buttons on the right to create, edit or delete MS Schedule+7.0 filters.

MS Schedule+7.0 Contacts Filters

○ <None Selected>

○ sched_contacts_1

New...

Edit...

Delete

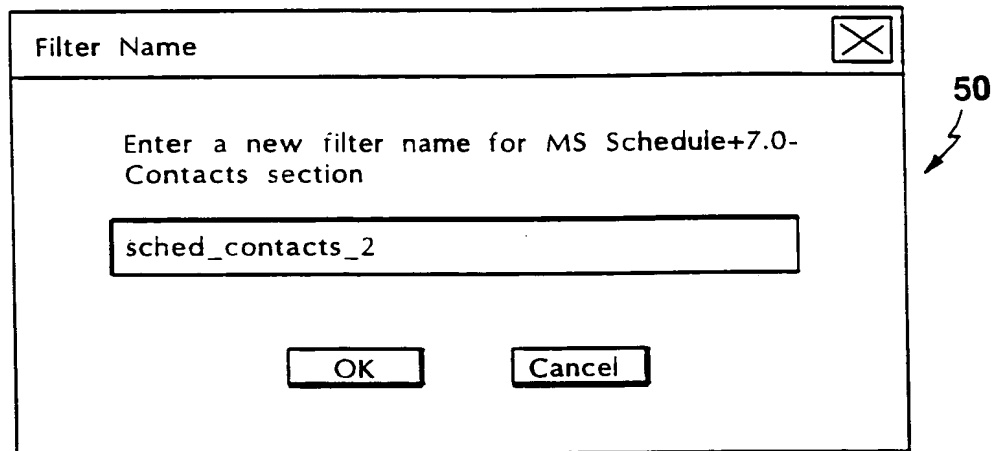OK   Cancel   Apply   Help   Field Mapping

FIG. 6

Filter Name ☒

Enter a new filter name for MS Schedule+7.0-
Contacts section

sched_contacts_2

OK   Cancel

50

FIG. 7

MS Schedule+ 7.0 filter - sched_contacts_2

| Conditions | Rules |

Add a new filter condition to the list or Update the currently selected filter condition (highlighted in the list box below). Only items passing the conditions below will be transferred.

Field ▼   66a

66c Operator: **66d**   **66e** Value:

Last Name   starts with ▶   M

66b

Add to List   Remove   Update

66f

List of Filter Conditions:

[Last Name] starts with 'M'

OK   Cancel   Help

66g

FIG. 8

MS Schedule+ 7.0 filter - sched_contacts_2

Conditions | **Rules**

Select the rule for combining the conditions. You can either specify that all conditions must be met in order for an item to be transferred, or that at least one of the conditions must pass.

⦿ All conditions must be met
◯ One or more conditions must be met

List of Filter Conditions:

[Last Name] starts with 'M'

OK | Cancel | Help

64

64a

FIG. 9

Filter language specification

Expression = Condition1 [AND Condition2] ... [OR Condition3] ...

Condition = ARG1 OP ARG2

OP = OP_SET_1 | OP_SET_2 | OP_SET_3 | OP_SET_4 | OP_SET_5 | OP_SET_6

OP_SET_1 = EQ | LE | GE | NE | LT | GT

OP_SET_2 = OP_SET_1 TODAY - | OP_SET_1 TODAY +

OP_SET_3 = OP_SET_1 NOW - | OP_SET_1 NOW +

OP_SET_4 = STARTS_WITH | CONTAINS | DOES_NOT_CONTAIN | IS_EMPTY |
          IS_NOT_EMPTY

OP_SET_5 = + | - | * | / | %

OP_SET_6 = IS

For Dates - ARG1 OP ARG2:
[Date Fieldname]   OP_SET_1   'YYYYMMDD' | [Date Fieldname2] | TODAY
[Date Fieldname]   OP_SET_2   integer

For Times - ARG1 OP ARG2
[Time Fieldname]   OP_SET_1   'HHMM' | [Time Fieldname2] | NOW
[Time Fieldname]   OP_SET_3   integer

For TextStrings - ARG1 OP ARG2
[String Fieldname]   OP_SET_1   'textstring' | [String Fieldname2]
[String Fieldname]   OP_SET_4   'textstring'

For Booleans - ARG1 OP ARG2
[Boolean Fieldname]   OP_SET_6   TRUE
[Boolean Fieldname]   OP_SET_6   FALSE

For Numbers - ARG1 OP ARG2
[Number Fieldname]   OP_SET_1   integer | float   **FIG. 10**
[Number Fieldname]   OP_SET_5   integer | float

200.   FOR each Record in history file
201.   Load record
202.   Write record to Workspace
203.   Next

FIG. 11

300. IF Use_Filter = TRUE and R_Application_Is_Filtering = FALSE THEN
301.   FOR each Record in the remote database
302.   Load record
303.   Filter the loaded record
304.   IF record passes the filter THEN mark as PASSED_FILTER
305.   ELSE mark as FAILED_FILTER
306.   Send record to synchronizer
307.   In Synchronizer: Write record to Workspace
308.   Next
309. ELSE IF Use_Filter = TRUE and R_Application_Is_Filtering = TRUE THEN
310.   Send the filter expression to R_Application
311.   Load filtered records
312.   IF the record passes current filter THEN Mark as PASSED_FILTER ELSE Mark as
    FAILED_FILTER
313.   Send records to synchronizer
314.   In Synchronizer: Write records to Workspace
315. END IF

FIG. 12

350. Form all records in the workspace into CIGs
351. For each CIG
352.   Compare the records in CIG
353.   Determine synchronization outcome
354.   IF a synchronization outcome is a conflict THEN
355.   IF one of the database records in the CIG does not pass the current filter, THEN skip CIG and mark results as DO NOT UPDATE any of the records
        ELSE resolve conflict by reference to a user-selected rule or input from the user
356.   END IF
357.   IF the most up to date record fails the filter, THEN mark all records as having failed the current filter
358.   IF the filter expressions contains an unmapped field and one of the database records in the CIG are marked as having failed the filter, THEN mark all records as having failed the filter
359.   IF a fanned out recurring record is partially outside of the current filter, THEN mark the record to be fanned when being unloaded and delete previous fanned nonrecurring records
360.
361. Next

FIG. 13

400. FOR each remote database record
401.   IF Use_Filter = TRUE and the filter is a static filter THEN
402.   IF record is marked as FAILED_FILTER THEN
403.   Delete record on the remote database
404.   Else IF the record is marked as PASSED_FILTER THEN add, delete, or modify
        record according to results of synchronization obtained during CAAR analysis
405.   ELSE IF Use_Filter = TRUE and the filter is a dynamic filter THEN
406.   IF record fails the current filter THEN
407.   Delete record on the remote database
408.   Else IF the record passes the current filter THEN add, delete, or modify record
        according to results of synchronization obtained during CAAR analysis
409.   END IF
410. Next

FIG. 14

450. FOR each local database record
451. IF Use_Filter = TRUE and the filter is a static filter THEN
452. IF record is marked as FAILED_FILTER THEN
453. IF CAAR outcome is to modify the record then modify the record on the local database
454. Else IF the record is marked as PASSED_FILTER THEN add, delete, or modify record according to results of synchronization obtained during CAAR analysis
455. ELSE IF Use_Filter = TRUE and the filter is a dynamic filter THEN
456. IF record fails the current filter but marked as PASSED_FILTER THEN
457. IF CAAR outcome is to modify the record then modify the record on the local database
458. Else IF the record passes the current filter THEN add, delete, or modify record according to results of synchronization obtained during CAAR analysis
459. END IF
460. Next

FIG. 15

FIG. 16